Data Reduction Effort at BNL

Brett Viren

Physics Department

BROOKHAVEN
NATIONAL LABORATORY

Local EDG
2017 Feb 6

Data Reduction Job and Efficacy

Job will consist of these major parts

→ gives reduction factor relative to canonical 12 bit packing and $4\times$ compression if saved after given step

1. **input unpacking:** with DAQ's raw data access library (DAL)
   → undo $\sim 4\times$ compression, $12 \rightarrow 32$ bit: $\sim 10\times$ **inflation**
2. **ADC mitigation:** stuck codes and non-linearity corrections
   → may improve compression at few 10% level (**total guess**)
3. **noise filter:** reduce excess noise
   → if pD/SP is noisy $4\times \rightarrow\sim 6\times$ compression, o.w. no data reduction here.
4. **signal processing:** waveform deconvolution + signal-ROI
   → $150\times$ (drop noise waveforms, reduce entropy)
5. **rebinning:** exploit oversampling but respect Nyquist
   → rebin-3 $400\times$ ($3\times$ and some entropy reduction)
6. **output packing:** save in raw-like format
   - needs a "reduced DAL" similar to "raw DAL"
   - relies on ROOT compression
   - this step is assumed for above reduction estimates

DocDB 2089 for details

Many Contexts for Data Reduction Code

Need steps 2 – 4 (ADC, noise filter, signal processing) for:

OM online monitoring inside artDAQ

- This is a "maybe"
- I think all OM should stick with showing raw quantities.

DQM aka prompt processing, aka p3s

- at least any reco job will need the "cooked" data

Offline production processing, reconstruction

- Either offline uses ∼10 TB reduced data (preferred) or
- Offline has to run steps 2 – 4 directly on 2.5 PB

→ if so, ∼10 TB reduced data is (hopefully) a side-product

Sim simulation connection to the reduction code:

- model ADC features and expected excess and inherent noise to generate data to **evaluate reduction efficacy**.

→ requires to pack simulation output in raw DAL format

- run signal processing on simulation output in any case (even if simulating noise-free and "perfect" ADC).

LArSoft and Wire Cell Toolkit

Right now:

- 35t ADC mit. and noise filt. **dev** in LArSoft (David)
- MicroBooNE noise filt **done**(?) in Wire Cell Toolkit (Xin)
  - → noise filt. is integrated to LArSoft (Brian & Brett)
  - → not following Service pattern due to lack of LS expertise
- Signal processing **dev** in WC prototype (Xin)
- Drift simulation **dev** in Wire Cell Toolkit (Brett)

Want/expect for protoDUNE:

- ADC mit. **dev** based on CE testing results
  - Likely in LS(?) Not strictly needed in WCT.
- Generalize MicroBooNE noise. filt. in WCT
  - Improve/generalize current WCT/LS integration.
  - This may end up being a no-op, but we have to be ready
- Implement sig. proc. in WCT and integrate into LS
- Finish drift sim in WCT and integrate into LS

Constraint: WCT will not depend on LS, integration involves writing LS job **modules** or **services** which call out to WCT.

What I can do and what I need help with

My software plate:

- Provide drift simulation and integrate uBoone and Milind's noise models.
- Do more data reduction studies using uBoone data?
- Sig.proc. prototype $\rightarrow$ WCT (either assist or do the work)
- Raw and reduced "DAL".
- Make LS integration succeed.

Where I would need help: (basically, LS integration)

- Working in LS, build, developer environment, configuration, release issues.
  - $\rightarrow$ this may have just gotten a little easier with the announcement of official Ubuntu 16.04 support by FNAL.
- Defining Service Interfaces for integration
  - need's coordination with David and greater LS community.
  - if we even go this route, module-level integration is an option

Other concerns

- Data reduction “keep up” needs $\sim$3000 cores
  - need $\sim 6\times$ this for full processing
- Data reduction “keep up” needs job management
  - a second instance of **p3s** would work.
  - need production processing or some group to include it in their scope.
- Data access library (DAL)
  - Must work closely with DAQ group
  - Want both packing (for sim) and unpacking code for all raw fragments
  - Want a “DAL” for reduced data (or include in raw DAL)
  - Must release new raw DAL **before** production DAQ writes new blocks. (ideally!)
  - New DAL code must still pack/unpack old data.

backups

LS Integration: Module vs. Service

- An art/LS job is a string of Modules communicating by data products.
- LS Services allow to share code across Modules or even outside of a Module.
- Defining a Service requires defining its method interface and the data types the methods take and return.
- The interface should be general enough but capture the abstraction. It's hard to get this right. Needs community involvement.
- Implementing an interface is somewhat more work than dumping code into a job module.
- Only one instance of an Art Service implementation can be created. Would like to extend Art to include "Tools" which overcome this limitation.

N.B.: Wire Cell Toolkit is heavily based on interfaces of which most are "Tools" rather than singleton "Services".